Linear Algebra Assignment 4 Report

110062219 翁君牧

January 6, 2023

Contents

1 Spam Emails I received

I selected 5 spam emails recently received from DockerCon, Slido, UberEATS, and other unknowns. They were submitted to the Google Form, saved to the comma-separated values file `A4_110062219.csv` and marked to be in folder $[0, 5)$ evenly.

2 Term Explanations of Text Mining

Below are the explanations of some terms regrading NLP *(Natural Language Processing)*.

2.1 TF-IDF

Term frequency - inverse document frequency is the product of **frequency** of a term and its **inverse document frequency**. Obviously **term frequency** is the ratio of total occurrence to total number of words. **Inverse document frequency** is defined as the natural logarithm of the ratio of number of documents to number of document containing the term.

In the viewpoint of this factor, the weight of a term is proportional to its frequency, whereas its specificity is reduced logarithmically by the number of documents it occurs.

2.2 Stop Words

There are some words in our common languages that have little meaning. For instance, articles such as 'a', 'the' and prepositions such as 'in', 'on' and other **function words** such as 'which', 'where', or **lexical words**.

3 Different Training/Testing Sets

There are 5 folders in the Kaggle data sets. Since we need select one to be the testing set while the lefts being the training set, there would be 5 combinations.

Table 1

Testing Set #	TP	FP	FN	TN	Accuracy	Precision	Recall
0	202	0	67	870	0.94	1.00	0.75
1	220	0	68	877	0.94	1.00	0.76
2	199	1	70	877	0.94	0.99	0.74
3	194	0	71	871	0.93	1.00	0.73
4	208	1	74	861	0.93	0.99	0.73

The elapsed time of each one is approximately 20 seconds on a full node of **Taiwania 3** with 56 cores.

In statistics, not only **accuracy**, **precision** and **recall** also really count. For instance, most of emails daily might be ham. So if we have a model that always claims an incoming email to be ham, it's likely it would have high accuracy. Nevertheless, it's totally useless. Therefore, this is why **recall** matters.

4 Distance Computing via SVD

So as to compute the distance of a vector $\mathbf{v}$ to the column space of matrix A via its SVD $U\Sigma V^H$, first we have the rank of a matrix to be the number of non-zero entries of Σ. Then we construct the orthonormal basis $\mathbf{b}$ of a matrix by the first $\text{rank}(A)$ of U. The projection $\mathbf{p}$ is $\mathbf{b} \cdot (\mathbf{b}^T \cdot \mathbf{v})$ thereby. So the residual $\mathbf{r}$ is then $\mathbf{p} - \mathbf{v}$. Thus the desired distance is the norm of the residual.

Note that since we have plenty of emails, i.e., vectors, we adopt matrix operations in practical. Moreover, we only need perform SVD of S, H once for spam and ham testing sets.

By the message[1] posted by the classmate on EEClass, I learnt that there's an argument of `numpy.linalg.svd()`, `full_matrices`, which could further accelerate the efficiency about twice.

4.1 Difference Between Least Squares and SVD

Another problem we're concerned is that why the results of `numpy.linalg.lstsq()` and `numpy.linalg.svd()` differs and which one is more accurate. I tend to believe that it's since that when performing least squares, we need solve the normal equation and find the Garmian matrix A^TA, which is likely to introduce inevitable floating point errors during the numerical computations. Nonetheless, in our case, least square outperformed SVD a bit.

5 Lower Rank Approximation and Latent Semantic Analysis

One of the most important application of SVD is to perform matrix approximation with lower rank, which has something to do with LSA.

[1] `https://eeclass.nthu.edu.tw/course/courseDiscuss/85854`

5.1 Different LRA

Our main goal is to reduce the **FN**, while keeping **FP** as little as possible. We could found that in general, the less the rank(H), the less the **FN** yet the more the **FP**; the more the rank(S), the less the **FN** yet the more the **FP**. The 3D graph of **FN** is illustrated in Figure 1.

Below are the two pairs that give rise to overall optimal result:

Table 2

rank(S)	rank(H)	TP	FP	FN	TN	Accuracy	Precision	Recall
200	300	272	2	10	860	0.99	0.99	0.96
600	600	274	4	8	858	0.99	0.99	0.97

Initially, I didn't feel any difference of elapsed time among the combinations of rank(S) and rank(H). After measuring the exact time, we could found that the less the rank, a bit less the time it consumed.

5.2 What's LSA?

In NLP, LSA is based on the distributional hypothesis, "linguistic items under similar distribution would tend to share the same semantic". As a consequence, we could reduce the size of our vocabulary while maintaining similar structure among documents.

There are several benefits of LSA. For instance, our original term-document matrix may contain some noisy. By means of LSA, we could get rid of these undesired noisy. In addition, the original matrix may be too sparse so that we would like to discard some terms that aren't occurred in every documents. Lastly, smaller matrix could be operated more effective.

5.3 How to Tune the Best Ranks?

So how could we tune the best ranks of S and H? It's just like what I did in the first subsection — brute-force.

I believe that what these ranks to be is what the parameters in the machine learning model to be. We have the residual – the loss function – and our objective is to minimize it.

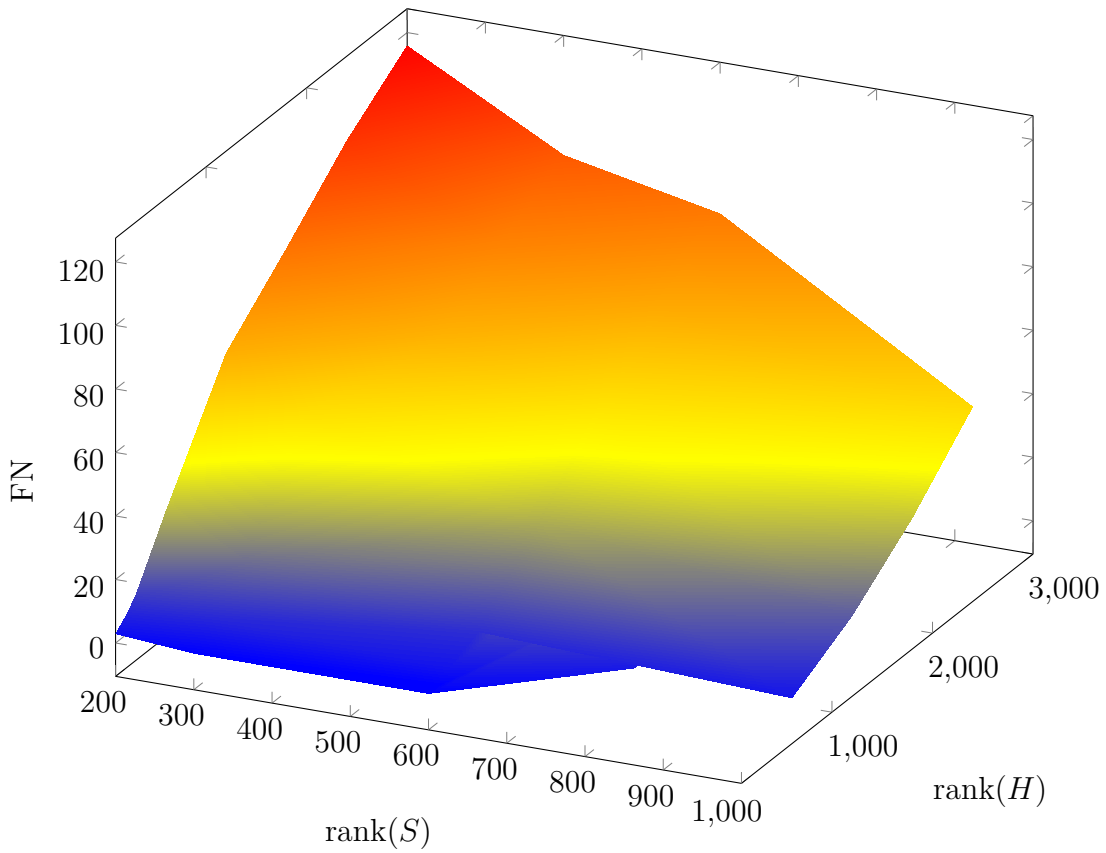

Figure 1: rank(S), rank(H) and FN

6 Distance Computing via NMF

If we would like to compute the distance between a vector to the column space of a matrix A via NMF WH, we could find that W itself form a orthonormal basis. As a consequence, the remaining part is the same as SVD.

In Python, Sci-Kit Learn package provides `sklearn.decomposition.NMF` model. Nevertheless, it requires quite a lot of time to perform the factorization. The default coordinate descent solver seemed not to utilize all CPU cores well. Though the multiplicative update solver worked better, it still took far slow than SVD. There are several initialization methods. `nndsvd` is quite faster than `nndsvda` and `nndsvdar` but there are warning about using `nndsvd` with multiplicative update.

In addition, we could observe that the number of components play a significant role in the wall time, whereas the rank of SVD impact the time slightly. What's worse, the results of NMF are really poor.

Acknowledgements

I thank to National Center for High-performance Computing *(NCHC)* for providing computational and storage resources.